MULTIMEDIA UNIVERSITY

STUDENT ID NO

MULTIMEDIA UNIVERSITY

FINAL EXAMINATION

TRIMESTER 2, 2016/2017

DCS5038 – PROGRAM DESIGN

(DIT & DBIS Groups)

27 FEBRUARY 2017
2.30 p.m. – 4.30 p.m.
(2 Hours)

INSTRUCTIONS TO STUDENTS:

1. This question paper consists of 9 pages with 5 questions.
2. **SECTION A**: Answer **ALL** questions.
3. **SECTION B**: Answer **ONLY ONE (1)** question.
4. Please print all your answers in the Answer Booklet provided.

SECTION A (75 Marks)

Instruction: *Answer **ALL** questions from this section and write your answers in the answer booklet provided.*

QUESTION 1 (25 Marks)

In value investing, high *return on equity (ROE)* and low *price to earnings (PE)* ratio are among the two financial indicators that can be used to shortlist high performance companies which are trading at a discount to their intrinsic values.

The formula for *ROE* and *PE* are as follows:-

$$ROE = \frac{earnings}{equity}, \quad PE = \frac{market\ capitalization}{earnings}$$

Guidelines.

- Initialize all the required variables.
- Get the inputs of *company name*, *earning*, *equity* and *market capitalization* from the user.
- Calculate the *ROE* and *PE*.
    - If *ROE* is less than 15; assign *"mediocre"* status.
    - Else
        - If *PE* is more than 10; assign *"expansive"* status. (*nested if-else* statement)
        - Else, assign "*undervalued*" status.
- Display the *company name* and its *status*.
- Repeat the whole process 5 times using a *do-while loop* for different company.

a. Based on the description given above, draw the **flowchart**. [13 marks]

b. Based on the description given above, write the **pseudocode**. [12 marks]

QUESTION 2 (25 Marks)

a. Trace the output of the following code segment. [5 marks]

```
int x = -17;

do
{
      printf( "%d\n", x );
      x += 13;
}
while(x <= 45);
```

Continued...

b. Write the *nested if-else* statement that determines the discount that would be given to a first time home buyer depending on their *status* and *income* as indicated in the table below.

[8 marks]

Status	Income (RM)	Discount (%)
Married	Less than RM 4000	10
	Less than RM 7000	7
	Less than RM 10,000	5
	Else	0
Single	Less than RM 5,000	3
	Else	0
Else	Discount is set to 0 and program terminates	

c. Write a full program that determines the MRT fare from Kajang station to three selected destinations.

- In *main()*
  - Get the *payment mode* (int) and *destination code* (char) from user.
  - Call function *getfare()*, passing the *payment mode* and *destination code* as parameter.
  - Display the calculated *fare*.

- In function *getfare()*
  Set the *fare* based on the *payment mode* and *destination code* (refer to the table below).
  - Use *if-else* statement to determine the *payment method*.
    - Use *switch-case* statement to set the *fare* based on the *destination code*.
  - Return *fare* to *main()*.

Payment Mode \ Destination	B (Bukit Bintang)	U (Bandar Utama)	S (Sungai Buloh)
1 (Cash)	4.30	5.00	6.40
2 (Touch n Go)	3.80	4.30	5.50

[12 marks]

SAMPLE OUTPUT

```
Enter payment mode      : 1
Enter destination code : U
Total fare              : RM 5.00
```

```
Enter payment mode      : 2
Enter destination code : B
Total fare              : RM 3.80
```

Continued...

QUESTION 3 (25 Marks)

a. For questions (i) – (iii), using the declarations and initializations below, indicate whether each of statements from (i) – (iii) is **valid** or **invalid**. If the statement is valid, indicate what value is displayed or assigned. If the statement is invalid, explain why.

[6 marks]

Given:
```
float a[8]= {11.0, 12.3, 7.0, 3.5, 7.4, -3.5, 8.0, 9.5};
float *pt;
int b = 5;
```

i. `printf("%.2f\n", a[8]);`
ii. `printf("%.2f\n", a[b+2]);`
iii. `pt = &a[4]; printf("%.2f\n",*pt);`

b. Trace the output for the following program. [2 marks]

```
#include <stdio.h>
#include <string.h>
int main()
{
     char name[]= "Pikachu", yourname[]= "Joyeousli";
     int count;

     if (strcmp(name, yourname)==0)
     {
        printf("%s and", name);
        printf("%s", yourname);
     }
     else
     {
         printf("%s", name);
         strcpy(yourname, name);
         printf("\n## is %d\n", strlen(yourname));
     }

     return 0;
}
```

Continued...

c. Trace the output for the following program. [6 marks]

```
#include<stdio.h>
void fun (int*, int*);

void main()
{
    int a=2, b=100;
    printf("\na = %d, b = %d", a, b);
    fun(&a, &b);
    printf("\na = %d, b = %d", a, b);
}

void fun(int *p, int *q)
{
   int sharp;
   sharp=*p;
   *p=*q;
   *q=sharp;
    printf("\na = %d, b = %d",*p , *q);
}
```

d. Answer the following questions based on the given records.

Name	Age	Height (m)
Johanna Lee	23	1.70
Arianna Fisher	18	1.72
Gigi Lee	19	1.65
Amirul Ahmad	22	1.79
Walter Andrews	22	1.68
Ervina Mahmud	23	1.62
Corine Tan	19	1.78
Wong Heng Zan	20	1.66

i. Create a structure called *Player* with the 3 attributes given above. Set the attribute name to 30 characters. [2 marks]

ii. Create a structure variable array called *p1* in *main()* with 8 elements. [1 mark]

iii. Assume that the array *p1* in (ii). has been set with the data above, write the codes to:
- Determine how many players have a height that exceeds 1.7 meters.
- Display the total number of players who meet that criteria. [3 marks]

e. Write a C program to display and count the total number of words that are in the message.*txt*. The The content of the file and the output screen are shown in the next page. [5 marks]

Continued...

Content of *message.txt*
`Believe in yourself! Have faith in your abilities.`
SAMPLE OUTPUT
`Believe in yourself! Have faith in your abilities.` `The total number of words are 8`

SECTION B (25 Marks)

Instruction: *Choose and answer **ONLY ONE (1)** question from this section and write your answers in the answer booklet provided.*

QUESTION 1 (25 Marks)

This program prompts the user to enter student details and determine their evaluation results based on their mock interview score at a Career Fair. Awards (in prizes or certificates) will be distributed to students based on their mock interview score.

Declare a *structure* called *Interview_profile* that has *stud_id (string)*, *course (string)*, *award (string)*, *score (float)* (which is an array with 3 elements) and *average (float)* .

In function main() :

- Create a structure variable array *SI*, which has 5 elements.
- Using a *for loop* :
  - Get from the user, the *stud_id* and *course*.
  - Using another *for loop*, prompt and get 3 evaluation scores for each student. In this loop, calculate the total score.
  - Calculate the *average* score.
  - Call function *get_result(...)* and pass the *average* score as parameter to determine the award for the student. Store the return value in *award*.
  - Display the results (*stud_id* , *course, average* score and *award*) as shown in the sample output.
- Determine which student's average score is the highest by calling function *find_max_avg(.....)*, passing the array *SI*.

In function *get_result (...)* :

- Identify and return the *award* based on the following table.

Average	Award
8.00 – 10.00	"Cash prize RM50 + Certificate"
6.00 – 7.99	"Book Voucher + Certificate"
Below 6.00	"Certificate"

Continued...

In function *find_max_avg(...)* :

- Parameter: structure variable array *SI* which has 5 elements.
- Using a *for* loop, determine which student has the highest *average* score and display the variable array element's details as shown in the sample output.

SAMPLE OUTPUT

```
Enter stud id   : 105152
Enter course    : DIT
Enter score 1   : 5
Enter score 2   : 6
Enter score 3   : 7

---------------------------------
    Results of Student 1
---------------------------------
ID        : 105152
Course    : DIT
Average   : 6.00
Award     : Book Voucher + Certificate

Enter stud id   : 103231
Enter course    : DIT
Enter score 1   : 8
Enter score 2   : 6
Enter score 3   : 2

---------------------------------
    Results of Student 2
---------------------------------
ID        : 103231
Course    : DIT
Average   : 5.33
Award     : Certificate

Enter stud id   : 104922
Enter course    : DBA
Enter score 1   : 9
Enter score 2   : 8
Enter score 3   : 9

---------------------------------
    Results of Student 3
---------------------------------
ID        : 104922
Course    : DBA
Average   : 8.67
Award     : Cash prize RM50 + Certificate

Enter stud id   : 102933
Enter course    : DBA
Enter score 1   : 7
Enter score 2   : 8
Enter score 3   : 7

---------------------------------
    Results of Student 4
---------------------------------
ID        : 102933
Course    : DBA
```

Continued...

```
Average    : 7.33
Award      : Book Voucher + Certificate

Enter stud id    : 102322
Enter course     : DIT
Enter score 1    : 5
Enter score 2    : 4
Enter score 3    : 5

----------------------------------
    Results of Student 5
----------------------------------
ID         : 102322
Course     : DIT
Average    : 4.67
Award      : Certificate

----------------------------------
    Best Student (Based on Average)
----------------------------------
104922
DBA
8.67
```

Displayed by *find_max_avg()*

Continued...

QUESTION 2 (25 Marks)

Write a **complete C program** that determines whether a person is eligible to apply for a bank's new personal loan scheme based on their debt-to-income *(DTI)* ratio.

In *main( )* :

- Declare all necessary variables.
- Open the file *profiles.txt* for reading.
- Open the *results.txt* for writing.
- If the file *profiles.txt* doesn't exist, display "*The file cannot be opened!*", else use a *while* loop to read each record from the file *profiles.txt* until the end of file:
  - Read the *IC, totalDebt* and *grossIncome.*
  - Call *calculateDTI(...)*, passing the *totalDebt and grossIncome* as parameters. The function calculates the *dti* value and returns it.
  - Determine the eligibility *status* based on the returned *dti* value. Refer to the following table.

dti	**status**
Above 0.4	"Not Eligible"
0.3 to 0.4	"KIV"
Below 0.3	"Eligible"

  - Write the *IC, dti* and *status i*nto file *results.txt.*
  - Keep track of how many persons has ***"Not eligible", "KIV"*** and ***"Eligible"*** statuses.
- Display the summary as shown in the sample output.
- *Close* file *profiles.txt* and *results.txt.*

- In *calculateDTI(...)*:
  - Calculate the debt-to-income (DTI) ratio, divide *totalDebt* by the *grossIncome*.

$$DTI = \frac{total\ debt}{gross\ income}$$

  - Return the *dti* value.

Contents of *profiles.txt*

<ic> *<totalDebt>* *<grossIncome>*

```
801005-04-5811 5300 9500
790605-01-6966 1500 8090
780509-01-6741 8056 11000
680202-05-4763 7800 13000
720610-04-6781 7860 20055
```

Continued...

Contents of *results.txt*

```
801005-04-5811 0.56 Not Eligible
790605-01-6966 0.19 Eligible
780509-01-6741 0.73 Not Eligible
680202-05-4763 0.60 Not Eligible
720610-04-6781 0.39 KIV
```

SAMPLE OUTPUT

```
----------Summary Results ---------
Not Eligible - 3 person(s)
KIV - 1 person(s)
Eligible - 1 person(s)
```

End of Page.